RSAC 2025 Conference

Many Voices. One Community.

SESSION ID: TPV-T02

Repeatable Supply Chain Security Failures in Firmware Key Management

Alex Matrosov
CEO & Head of REsearch
binarly

Fabio Pagani
Vulnerability REsearch Lead
binarly

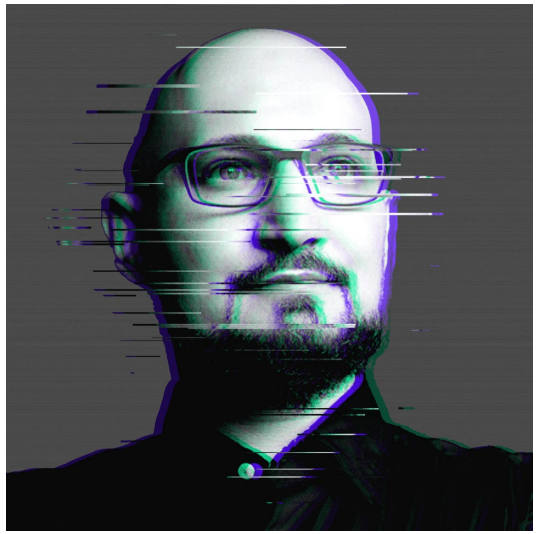
Disclaimer

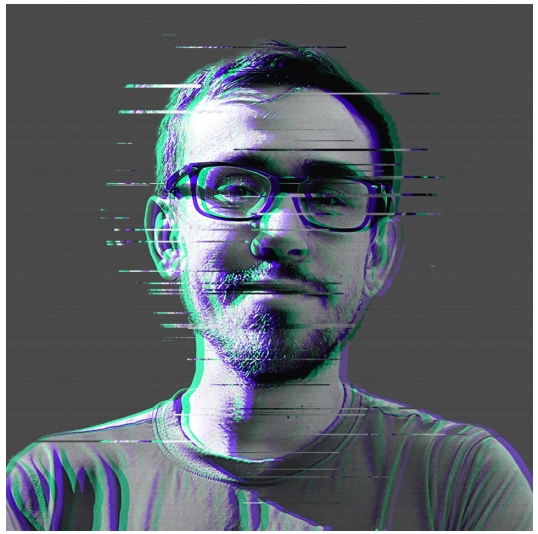
Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Binarly REsearch Team





Alex Matrosov
@matrosov

Fabio Pagani
@pagabuc

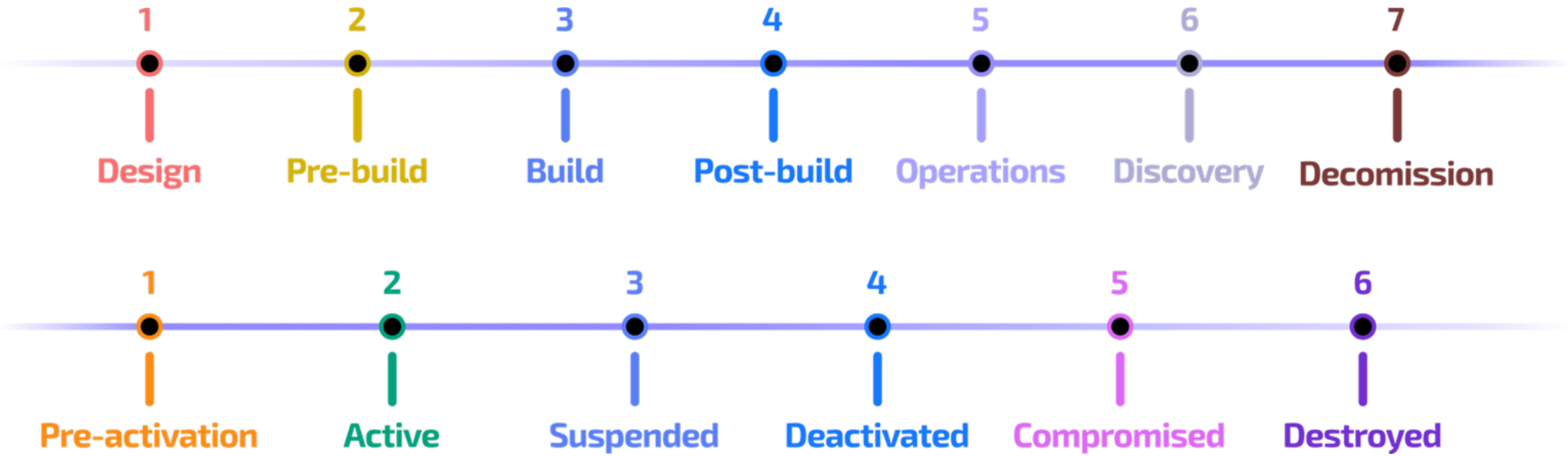
https://www.binarly.io/articles
https://www.binarly.io/advisories

binarly

RSAC™ | 2025 Conference

All this has happened before.
All this will happen again.

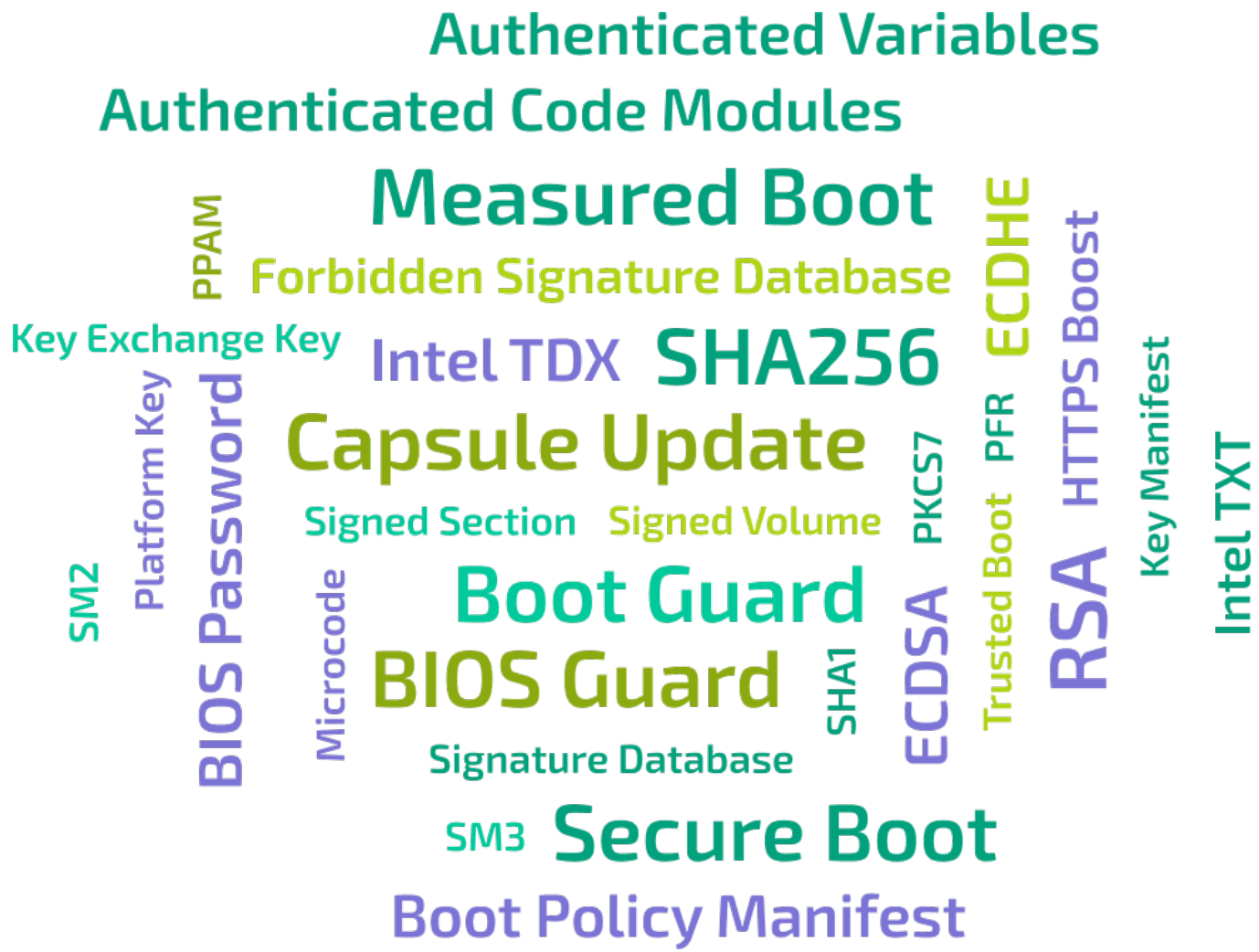
Cryptographic Key Management is Hard!



binarly

RSAC 2025 Conference

Introduction



Introduction

What can go wrong?

Security risks arising from firmware developer and device vendor breaches

2022

- Intel PPAM expired certificate
- LC/FC data breach

2023

- MSI OEM data breach
- Intel BootGuard key leakage impact

2024

- PKfail + BlackLotus Demo
- Supermicro

2025

- DBX inconsistency
- Intel BootGuard again?
- AMD Microcode validation is broken

binarly

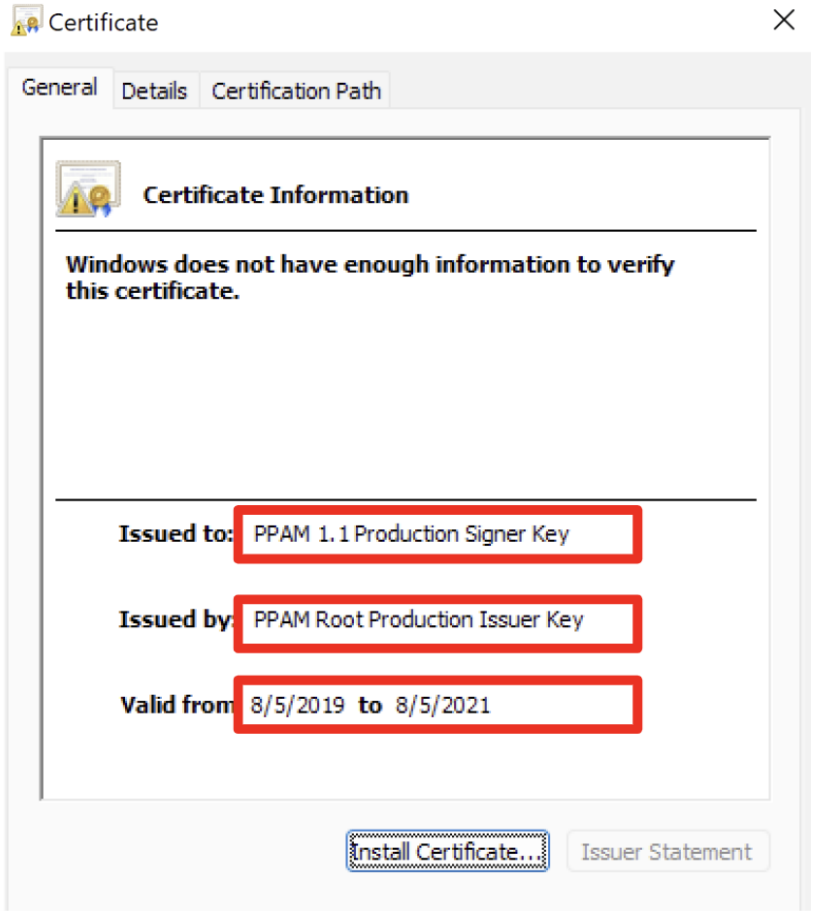
RSAC 2025 Conference

[2022] Intel Platform Properties Assessment Module (PPAM) Expired Certificate Story

[2022] Intel PPAM expired certificate

- Platform Properties Assessment Module (PPAM) measures the integrity of SMM code
- Binary signed by Intel that runs before System Management Mode (SMM) entry point
- PKCS7 certificate provides a digital signature for PPAM
- Multiple devices with expired PPAM certificate

https://www.binarly.io/blog/black-hat-2022-the-intel-ppam-attack-story



Revisiting Intel PPAM expired certificate

- Retrospective scan on our dataset revealed that 68% of certificates in-the-wild are expired
- We also found few recents devices deployed with PPAM debug certificates
- This is not a security vulnerability, but shows the bad security practices

```
Version: 3 (0x2)
Serial Number:
    63:00:33:3a:47:12:7f:a3:eb:ad:a1:61:ad:00:01:00:33:3a:47
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=US, ST=CA, L=Santa Clara, O=Intel Corporation,
        OU=SSG, CN=PPAM Root Debug Issuer Key
Validity
    Not Before: Jun 12 10:59:01 2019 GMT
    Not After : Jun 12 10:59:01 2020 GMT
Subject: C=US, ST=CA, L=Santa Clara, O=Intel Corporation,
         OU=SSG, CN=PPAM 1.1 Debug Signer Key
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
        Modulus:
            00:bf:ec:93:b2:59:0f:7f:ef:e1:cc:ae:bc:33:27:
            e5:34:e6:d8:eb:00:17:aa:51:65:56:74:e2:10:a5:
            19:dc:a1:89:74:ab:45:f1:0a:9a:5b:54:af:14:42:
...
```

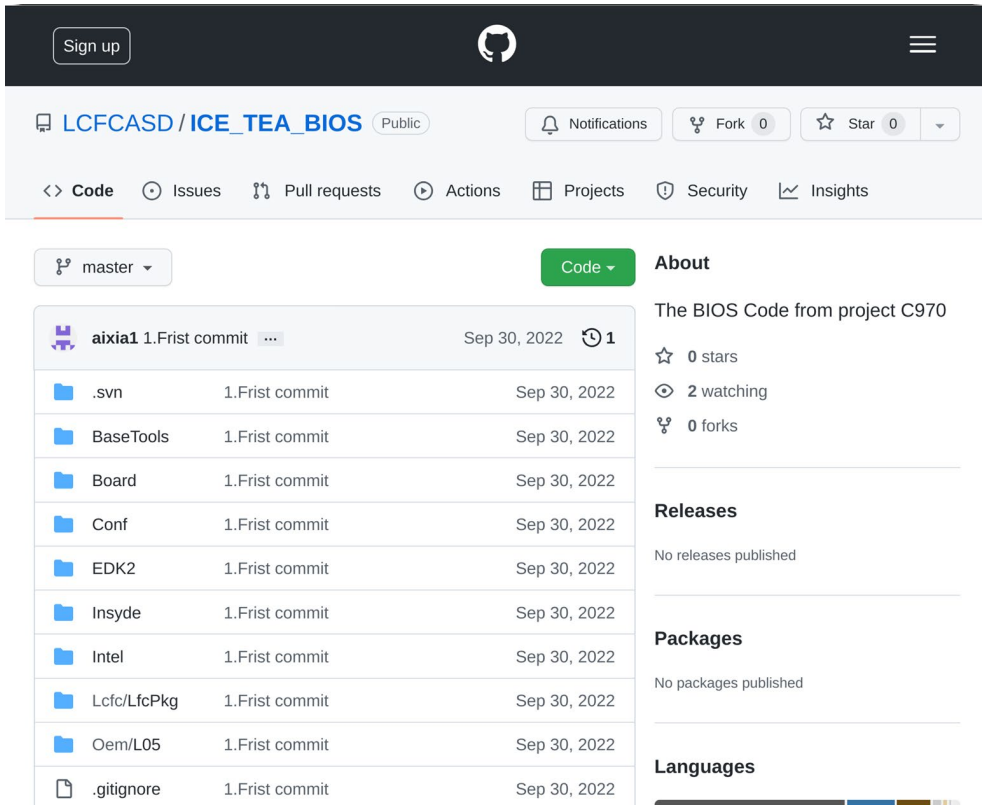
RSAC 2025 Conference

[2022] Lenovo LCFC OEM Data Breach and Leaked Keys

[2022] LC/FC data breach

- Alder Lake's UEFI firmware was leaked on GitHub
- Reference implementation (Intel), IBV solution (Insyde) and OEM implementation (Lenovo)
- 6GB of source code, binary blobs, debugging tools and multiple private keys

https://www.binarly.io/blog/leaked-intel-boot-guard-keys-what-happened-how-does-it-affect-the-software-supply-chain

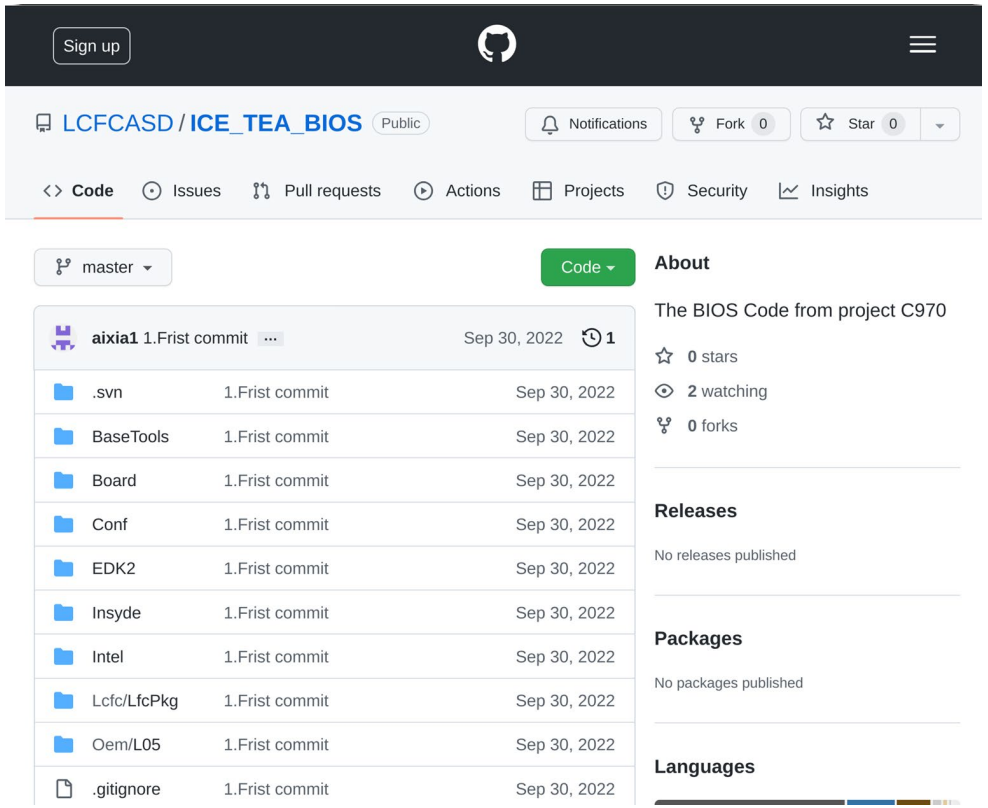


[2022] LC/FC data breach

Leaked private keys:

- Intel Integrated Sensors Hub (ISH) signing key
- Intel Boot Guard KM/BPM keys
  - Found on devices from Lenovo, Supermicro and Intel

https://www.binarly.io/blog/leaked-intel-boot-guard-keys-what-happened-how-does-it-affect-the-software-supply-chain



RSAC 2025 Conference

[2023] MSI OEM Data Breach and Leaked Keys

[2023] MSI OEM data breach

- Breach from the Money Message ransomware group
- 1.5TB of source code, production databases and multiple private keys

https://www.binarly.io/blog/leaked-msi-source-code-with-intel-oem-keys-how-does-this-affect-industry-wide-software-supply-chain

[2023] MSI OEM data breach

- Intel BootGuard BPM/KM keys
  - More than 100 MSI devices affected
- FW Image Signing Keys
  - Around 60 MSI devices affected
- Intel OEM Platform Key
  - Orange unlock: more powerful than Boot Guard key
  - Found on devices from HP, Lenovo, AOPEN, CompuLab, and Star Labs

https://www.binarly.io/blog/leaked-msi-source-code-with-intel-oem-keys-how-does-this-affect-industry-wide-software-supply-chain

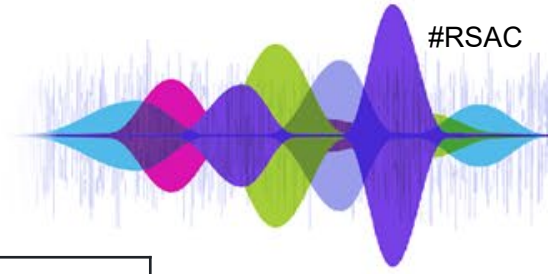
RSAC 2025 Conference

[2025] Clevo ODM Leaked Keys

[2025] Clevo Boot Guard keys leak

- Binarly was notified by Thierry Laurion about a possible leak of Boot Guard keys from Clevo in March 2025
- Firmware update package (400MB uncompressed size)
- Contains user manuals, internal tooling, firmware images and Boot Guard private keys

```
$ openssl rsa -text -in CreateDeleteBIOSKey.keyprivkey.pem
Private-Key: (3072 bit, 2 primes)
modulus:
    00:c5:81:81:14:d9:69:55:6c:38:a4:1a:f3:1c:a2:
    01:10:cf:02:f1:0c:73:f6:44:dc:e8:ae:25:69:6b:
    fa:14:ca:95:58:1a:d6:63:95:e4:97:57:a7:12:ea:
    eb:32:c8:b1:34:4b:1e:97:08:68:b9:7f:54:89:ba:
    09:86:cd:f1:1a:0d:e8:0d:18:38:e2:a0:bb:ad:87:
    d3:c2:3f:d5:e4:e8:4e:cd:e7:7d:d4:67:3b:33:ee:
    4a:ce:7c:aa:88:45:fa:ac:74:d1:a9:42:14:c7:1a:
    88:9c:cf:61:ef:b6:36:65:a7:2d:05:21:1e:a9:3a:
    fe:2d:09:09:0e:e7:e8:eb:e6:6e:61:95:11:a8:b5:
    78:b4:8c:0f:49:82:47:7b:87:b5:0d:a8:57:9f:16:
    12:8f:d8:ef:e6:84:49:f9:f7:37:a1:00:5f:4d:92:
    a9:e7:08:3c:bc:04:63:2f:94:49:1c:23:1f:72:dd:
    25:ed:bb:d1:92:69:11:2b:23:a4:72:02:89:e2:ab:
    93:e9:1f:e4:4a:f8:ac:bd:12:e7:69:3e:b9:a1:80:
    04:f8:2f:00:20:fd:15:12:2b:7d:f7:91:bc:33:84:
    bf:e1:e7:26:58:c3:00:29:02:f6:66:9e:69:68:f2:
    b3:ea:27:f5:b3:cf:f6:0b:1a:d3:28:82:63:ef:53:
    ab:e4:d8:dc:c6:57:a7:ff:9d:35:80:a8:c6:35:af:
    9d:4c:62:e4:9c:d3:db:e9:07:ad:8d:9c:8a:85:c6:
    50:24:29:8b:da:7e:90:24:70:cf:0e:b4:15:46:8e:
    89:cd:24:e6:c6:b4:42:0e:13:b3:1d:3d:f8:87:52:
    70:2e:18:53:26:64:35:ed:16:9c:cd:23:f5:58:2f:
...
```



[2025] Clevo leak - Impacted Devices

Firmware / Device name	ODM	IBV	Version	Release date
XPG Xenia 15G G2303_V1.0.8	Clevo	Insyde	6.2.8320.0	2023-06-14
Gigabyte G5 KE	Clevo	Insyde	FB05	2023-03-07
Gigabyte G5 KF 2024	Clevo	Insyde	FD06	2024-01-10
Gigabyte G5 KF5 2024	Clevo	Insyde	FD10	2024-12-09
Gigabyte G5 ME	Clevo	Insyde	FB04	2023-06-05
Gigabyte G5 MF	Clevo	Insyde	FB03	2023-04-14
Gigabyte G6 KF	Clevo	Insyde	FB06	2023-10-23
Gigabyte G6X 9KG 2024	Clevo	Insyde	FB10	2025-02-04
Gigabyte G7 KF	Clevo	Insyde	FB10	2024-02-16
NoteBook Firmware 1.07.07TRO1	Clevo	Insyde	6.2.8319.7	2023-09-05
NoteBook Firmware 1.07.09TRO1	Clevo	Insyde	6.2.8319.9	2023-11-28

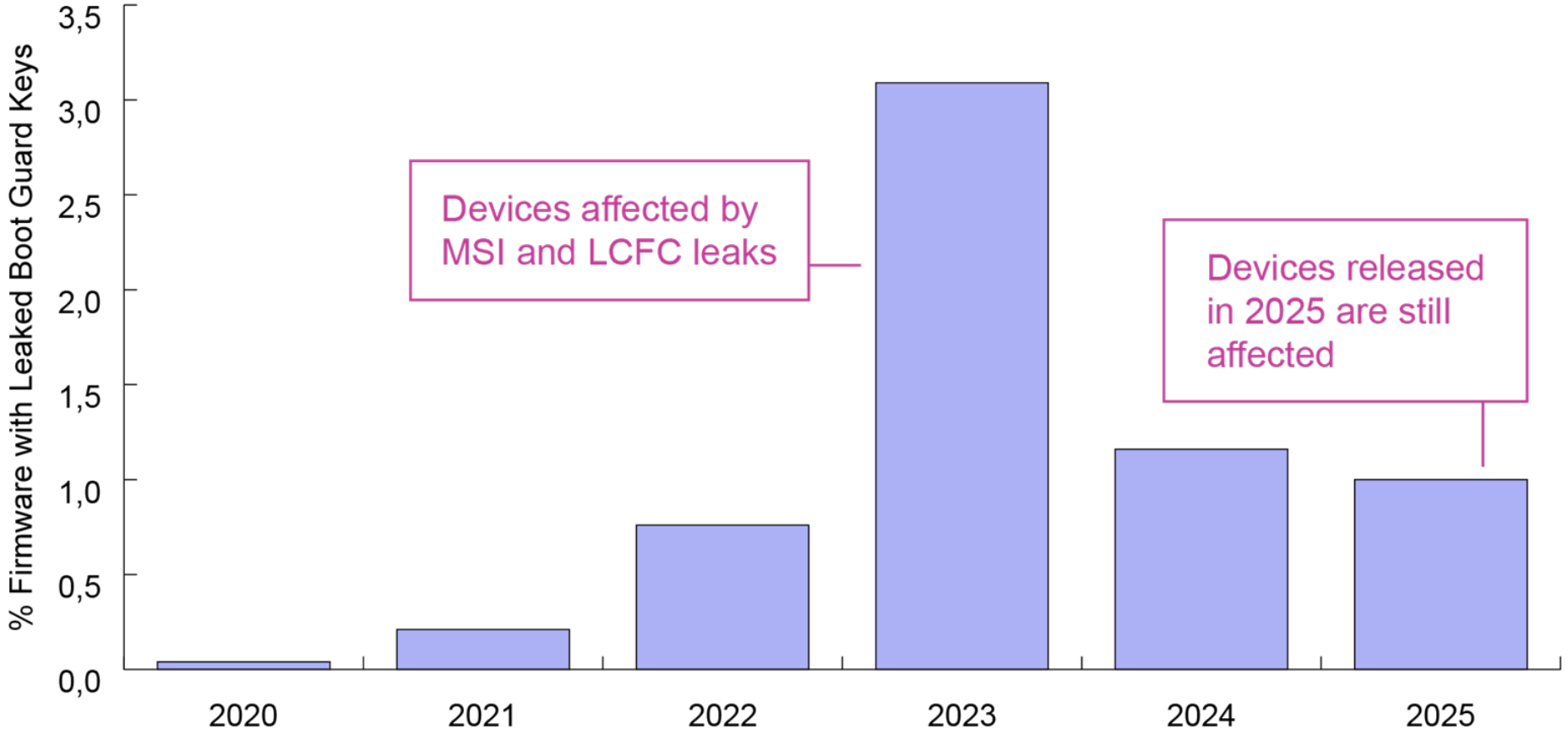
RSAC 2025 Conference

Intel Boot Guard
Impact of Leaked Keys

Boot Guard - Introduction

- Hardware-based technology intended to protect against execution of non-genuine UEFI firmware
- Multiple components and cryptographic keys involved:
  1. **Authenticated code module (ACM)**: Intel-signed code that runs before the firmware and cryptographically verifies the firmware
  2. **Key Manifest (KM)**: verifies Boot Policy Manifest
  3. **Boot Policy Manifest (BPM)**: verifies Initial Boot Block (basically, the firmware)

Impact of Boot Guard keys leakage



Impact of Boot Guard keys leakage

Why current devices are still vulnerable to a leak from years ago?

The Boot Guard Key Manifest hash is fused in the platform hardware and it **cannot** be changed!

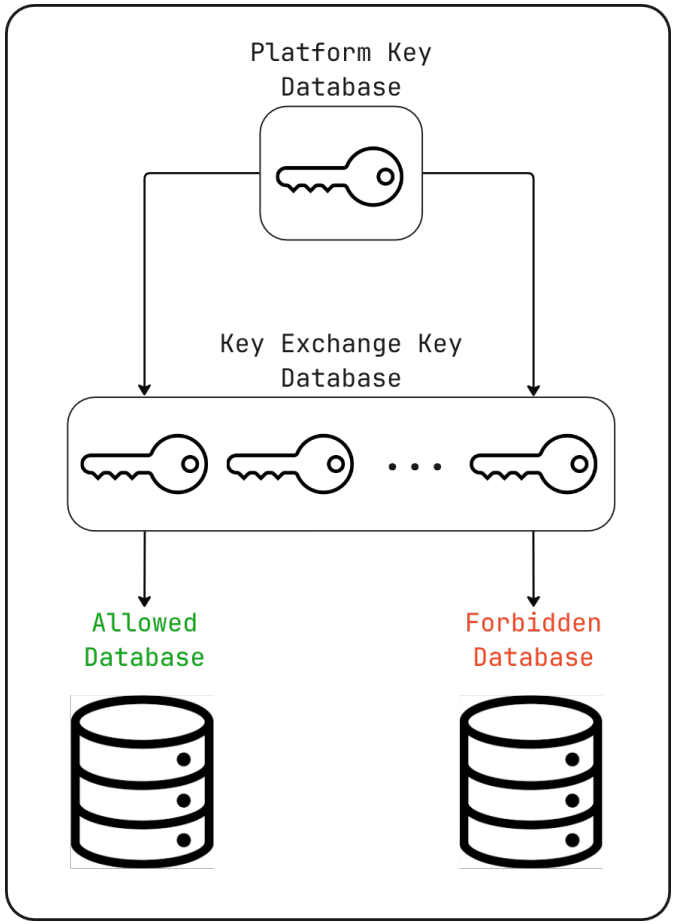
RSAC 2025 Conference

[2024] PKfail
Leaked Platform Key Story

Many Voices.
One Community.

UEFI Secure Boot

- Allows only trusted, digitally signed software to run during system startup, preventing malware and unauthorized code execution.
- Bypassing Secure Boot allows for bootkit and rootkit execution
- Four databases:
  - PK, KEK, db, dbx



[2024] PKFail

While adding support for Secure Boot to our Binarly Transparency Platform, we found an "*interesting*" Platform Key:

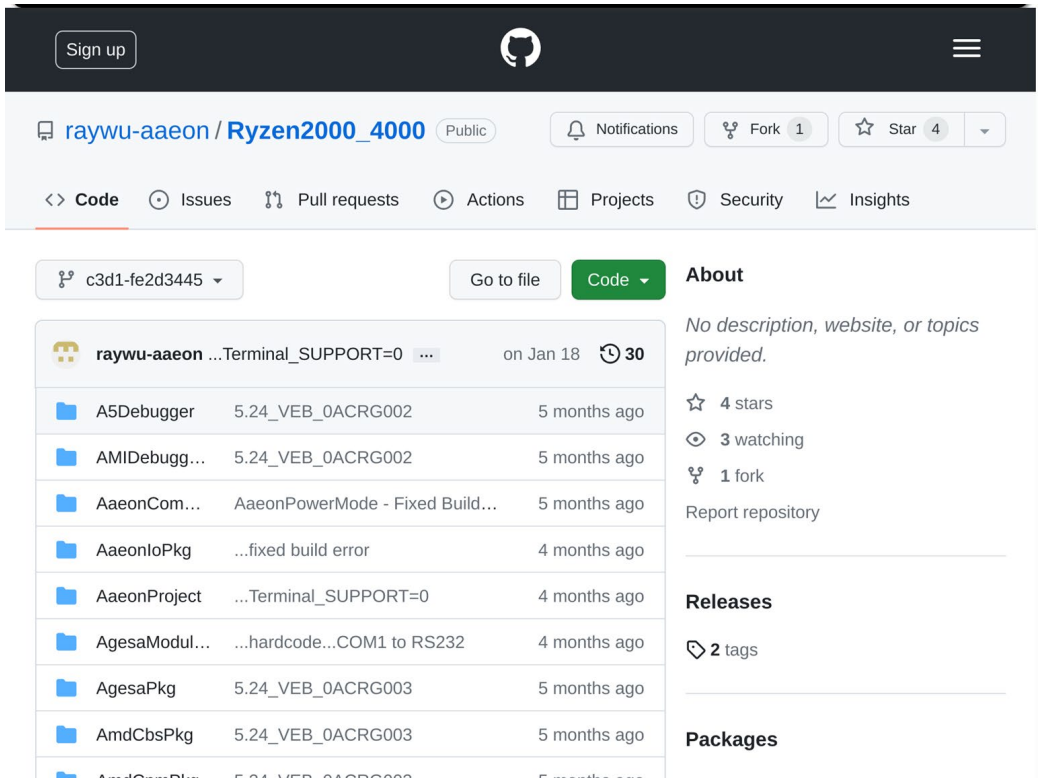
https://www.binarly.io/blog/pkfail-untrusted-platform-keys-undermine-secure-boot-on-uefi-ecosystem

```
Version: 3 (0x2)
Serial Number:
    55:fb:ef:87:81:23:00:84:47:17:0b:b3:cd:87:3a:f4
Signature Algorithm: sha256WithRSAEncryption
Issuer: CN=DO NOT TRUST - AMI Test PK
Validity
    Not Before: Nov  8 23:32:53 2017 GMT
    Not After : Nov  8 23:32:52 2021 GMT
Subject: CN=DO NOT TRUST - AMI Test PK
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
        Modulus:
            00:e7:36:7b:20:92:ba:7f:aa:a3:f6:0e:49:08:87:
            f5:1c:11:33:ba:5d:f8:9b:5c:ed:c7:90:e4:f3:41:
...
```

binarly

[2023] AAeon leak

- In January 2023, the repository **Ryzen2000_4000** is published on GitHub
- Contains IBV (AMI) reference implementation, ODM (AAeon) implementation and private keys
- Remained public until AMI sent a DMCA to GitHub in June 2023



[2023] AAeon leak

```
$ openssl x509 -noout -text -in FW_pubKey.cer  | rg "Issuer:|Subject:"
        Issuer: CN=DO NOT TRUST - AMI Test PK
        Subject: CN=DO NOT TRUST - AMI Test PK
```

```
$ openssl pkcs12 -in FW_priKey.pfx -nodes
Enter Import Password:
```

```
$ cat AmiTestKey.sdl | grep password -C3
TOKEN
        Name  = "FW_PFX_Password"
        Value  = "abcd"
        Help  = "Specifies the password to use when opening a PFX -
Private Key container file."
        TokenType = Expression
        TargetMAK = Yes
End
```

Oh, hi! I am a private key that's been available on GitHub for 6 months! 🤷‍♂️

binarly

RSAC 2025 Conference

Retrospective view on PKFail

Dataset with 80,000 UEFI firmware images:

- Spanning over 10 years
- Includes every major vendor (Lenovo, Dell, HP, Intel..)

Results:

- 10% of images use non-production keys
- 8% of images when selecting images released in the past 4 years
- 22 unique non-production keys identified

binarly

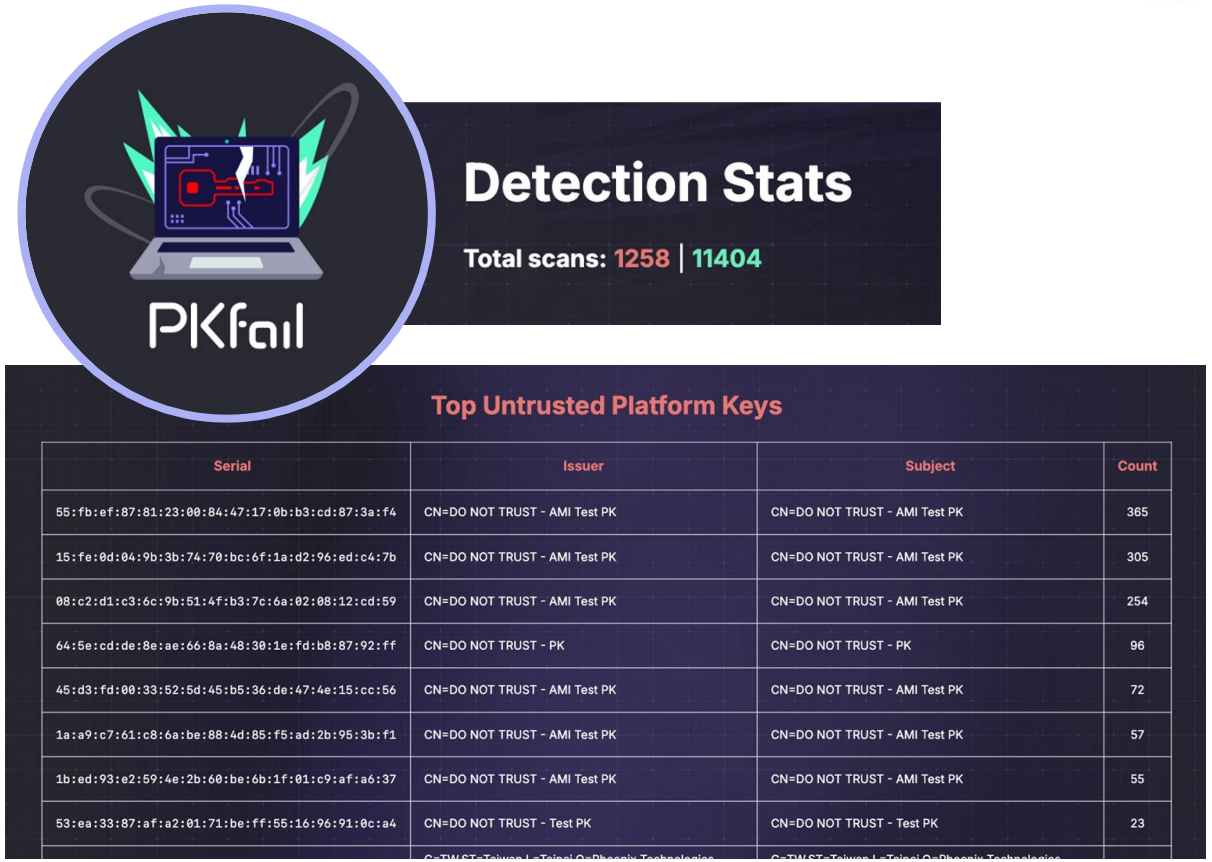
Retrospective view on PKFail

Certificate Serial Number	Certificate Subject	Certificate Issuer	Last Seen	First Seen	Products	Vendors
55:fb:ef:87:81:23:00:84:47:17:0b:b3:cd:87:3a:f4	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	2024-06	2018-04	364	Acer, Dell, Fujitsu, Gigabyte, Intel, Lenovo, Supermicro
-08:c2:d1:c3:6c:9b:51:4f:b3:7c:6a:02:08:12:cd:59	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	2024-06	2022-06	167	Acer, Dell, Gigabyte, Supermicro
-15:fe:0d:04:9b:3b:74:70:bc:6f:1a:d2:96:ed:c4:7b	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	2024-03	2015-01	483	Acer, Dell, Gigabyte, Intel, Lenovo, Supermicro
-1b:ed:93:e2:59:4e:2b:60:be:6b:1f:01:c9:af:a6:37	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	2023-01	2014-12	287	Dell, Fujitsu, Gigabyte, HP, Intel, Lenovo, Supermicro
1a:a9:c7:61:c8:6a:be:88:4d:85:f5:ad:2b:95:3b:f1	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	2021-03	2012-05	157	Acer, Dell, Fujitsu, Gigabyte, HP, Lenovo, Samsung, Supermicro

Binarly's pk.fail detection service

Binarly released a free detection service for the community on disclosure date:

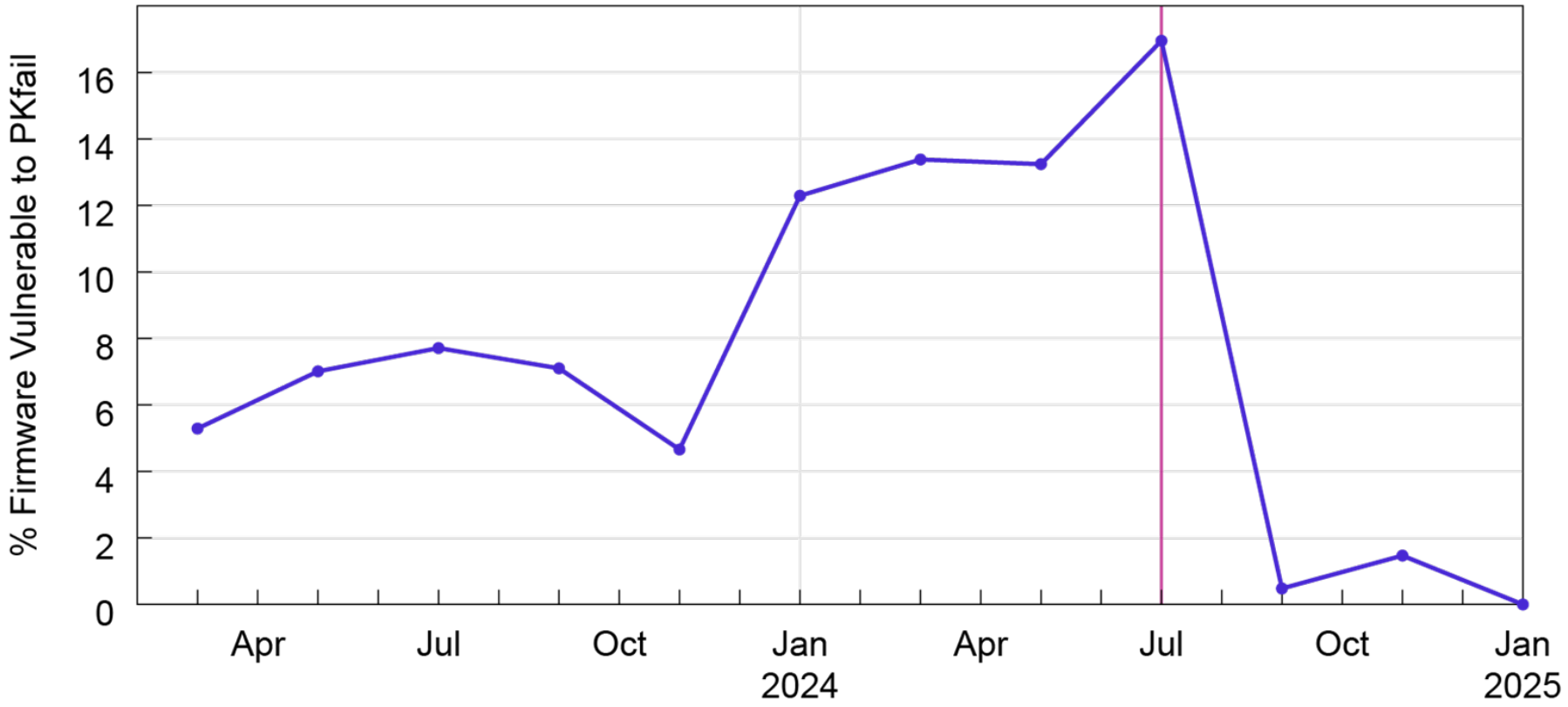
- Users uploaded **12,662** firmware images
- Found untrusted keys in **1,258** of them (9.94%)
- The most common key remains the leaked AMI key 🔥



Serial	Issuer	Subject	Count
55:fb:ef:87:81:23:00:84:47:17:0b:b3:cd:87:3a:f4	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	365
15:fe:0d:04:9b:3b:74:70:bc:6f:1a:d2:96:ed:c4:7b	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	305
08:c2:d1:c3:6c:9b:51:4f:b3:7c:6a:02:08:12:cd:59	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	254
64:5e:cd:de:8e:ae:66:8a:48:30:1e:fd:b8:87:92:ff	CN=DO NOT TRUST - PK	CN=DO NOT TRUST - PK	96
45:d3:fd:00:33:52:5d:45:b5:36:de:47:4e:15:cc:56	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	72
1a:a9:c7:61:c8:6a:be:88:4d:85:f5:ad:2b:95:3b:f1	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	57
1b:ed:93:e2:59:4e:2b:60:be:6b:1f:01:c9:af:a6:37	CN=DO NOT TRUST - AMI Test PK	CN=DO NOT TRUST - AMI Test PK	55
53:ea:33:87:af:a2:01:71:be:ff:55:16:96:91:0c:a4	CN=DO NOT TRUST - Test PK	CN=DO NOT TRUST - Test PK	23

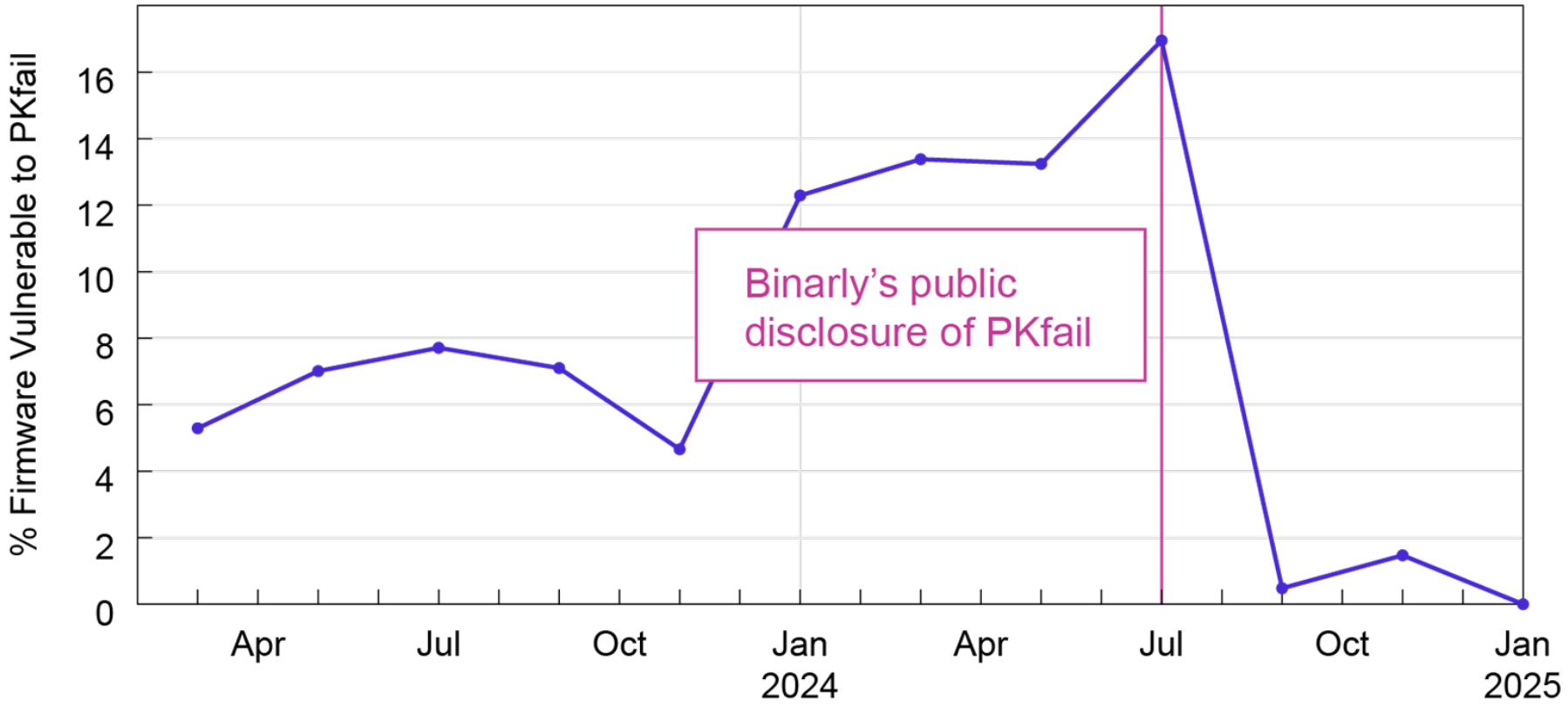
binarly

Impact of PKFail on the UEFI ecosystem



binarly

Impact of PKFail on the UEFI ecosystem



No firmware vulnerable to PKfail detected so far in 2025!

binarly

RSAC 2025 Conference

Distribution of PK across vendors

	2021	2022	2023	2024	Total (Unique)
Acer	4	3	1	3	6
Dell	18	22	16	17	28
Fujitsu	5	7	8	6	9
Gigabyte	6	10	12	11	15
HP	3	3	3	3	3
HPE	2	2	2	2	2
Intel	5	10	5	1	10
Lenovo	37	106	120	92	154
Msi	4	5	5	3	5
Supermicro	3	3	3	1	4

RSAC 2025 Conference

Many Voices.
One Community.

PKfail PoC

https://www.youtube.com/watch?v=SPl7zfC-CmQ

PKfail PoC (Linux)

https://www.youtube.com/watch?v=CveWt3gFQTE

RSAC 2025 Conference

[2024] Supermicro BMC Test Key Reuse

[2024] Supermicro test key

- Baseboard Management Controller (BMC) firmware also affected by a similar problem
- Test key was not leaked
- Despite our report, nothing changed:
  - Public key modulus still present in the latest firmware images
  - Parts of the image is still signed with the test key

```
Version: 3 (0x2)
Serial Number:
    1a:da:e6:cf:23:66:6a:36:d9:dd:69:4c:2f:ba:30:14:90:f7:3d:5e
Signature Algorithm: sha512WithRSAEncryption
Issuer: C = US, ST = CA, L = SanJose, O = Super Micro Computer Inc.,
        CN = RD1 BMC Test Key - DO NOT TRUST
Validity
    Not Before: Feb 14 03:14:28 2020 GMT
    Not After : Feb  1 03:14:28 2070 GMT
Subject: C = US, ST = CA, L = SanJose, O = Super Micro Computer
Inc.,
        CN = RD1 BMC Test Key - DO NOT TRUST
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (4096 bit)
        Modulus:
            00:c6:b3:42:c9:36:c3:a1:24:0c:ec:e5:1a:31:96:
            5b:1d:a6:c7:85:66:50:bf:59:78:9c:2d:8d:07:5e:
            6f:9b:f0:a0:70:7a:42:f0:0a:68:bd:e1:aa:80:ef:
            2c:70:bd:7a:36:59:6a:ca:2a:1d:21:f1:1c:a1:31:
...
```

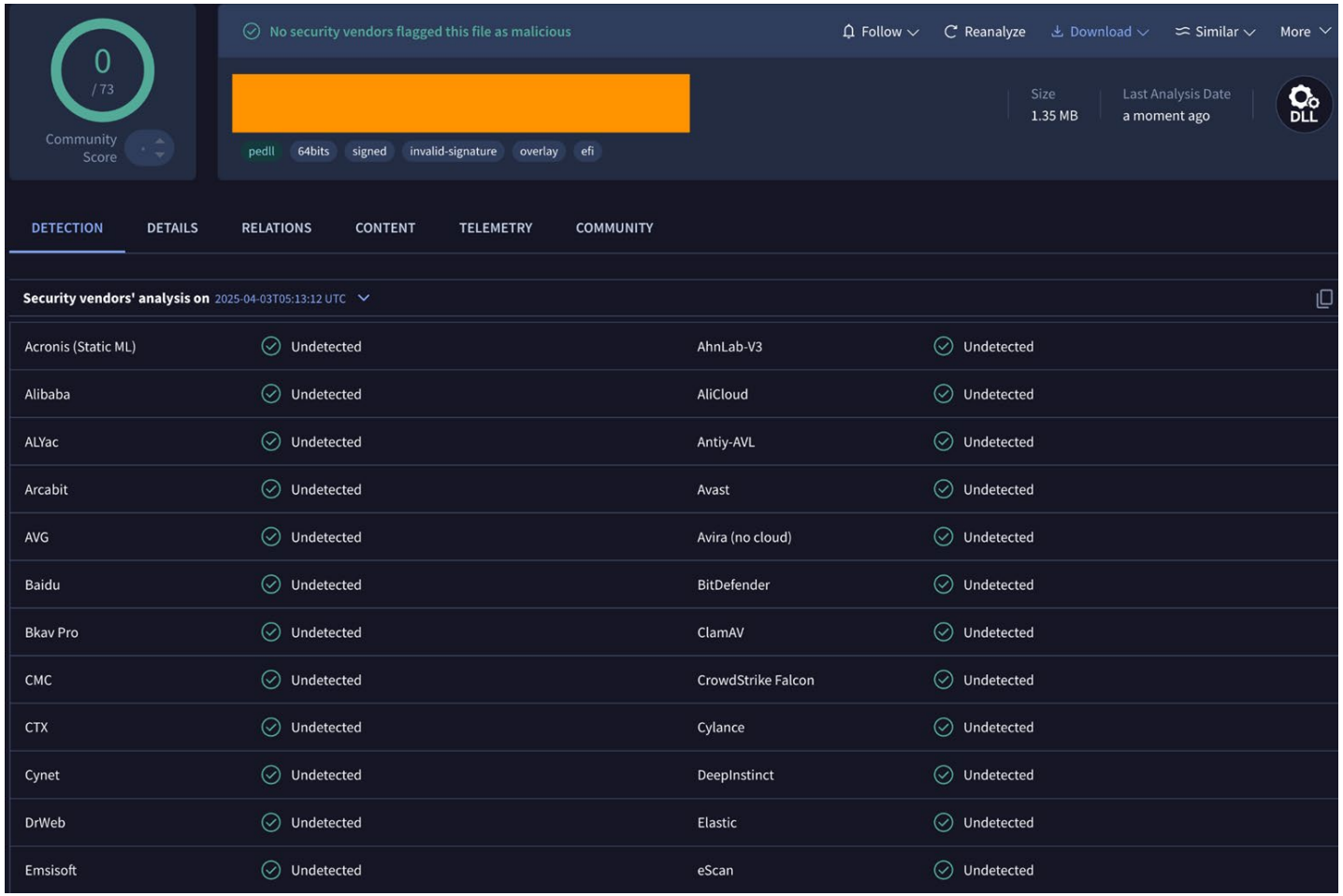
https://www.binarly.io/blog/repeatable-failures-test-keys-used-to-sign-production-software-again

binarly

RSAC 2025 Conference

[2025] Microsoft Signed UEFI Module Universal Secure Boot Bypass CVE-2025-3052 (BYOVD)

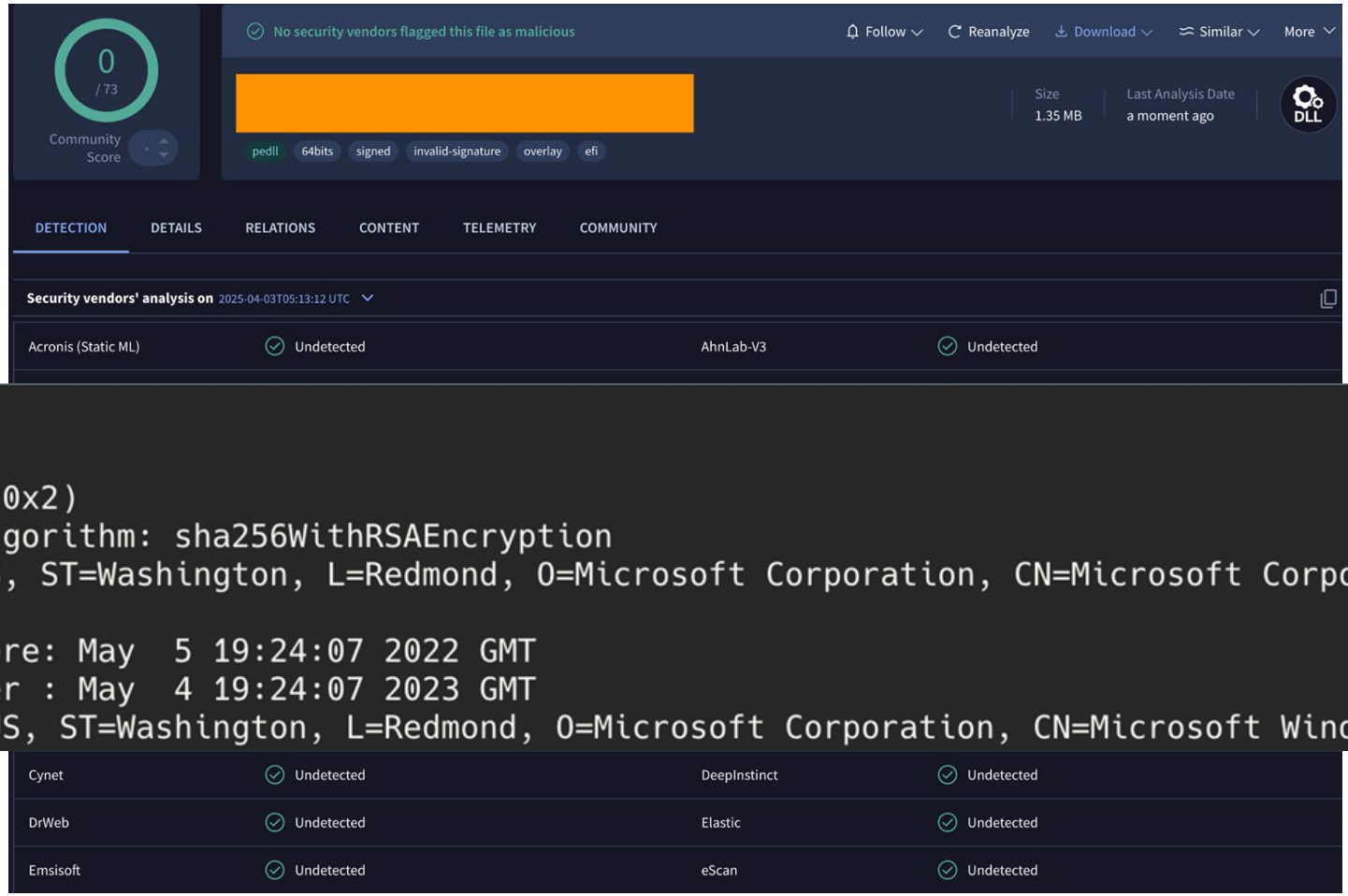
[2025] SignedModule.efi found on VT



binarly

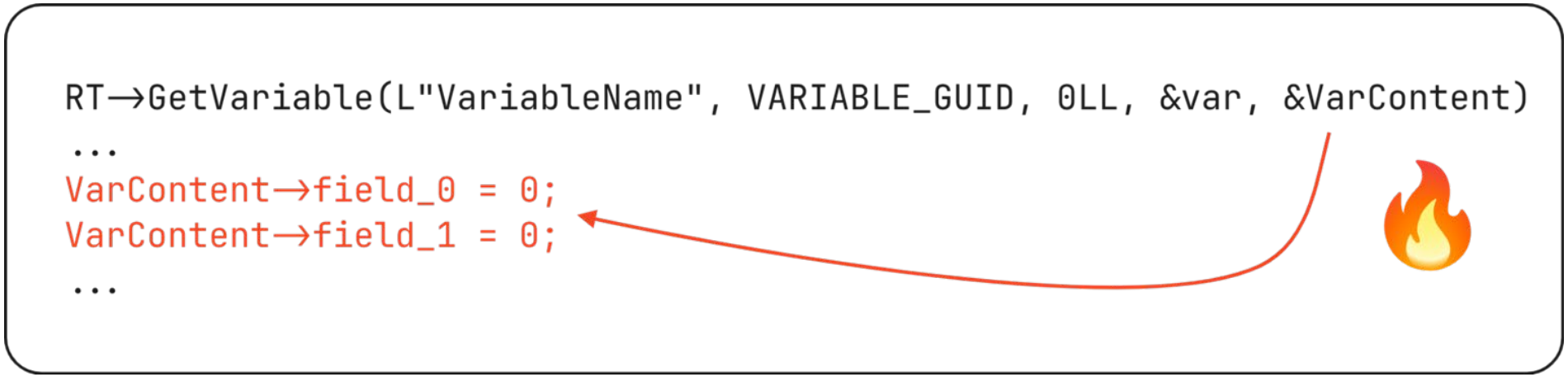
RSAC 2025 Conference

[2025] SignedModule.efi found on VT



[2025] SignedModule.efi found on VT

- Secure Boot is prone to BYOVD attacks
- Ongoing disclosure with CERT//CC (*stay tuned for more details…*)
- Module signed with “Microsoft Corporation UEFI CA 2011”, trusted by basically every device out there



RSAC 2025 Conference

DEMO
Proof of Concept for CVE-2025-3052

RSAC 2025 Conference

[2025] DBX Inconsistency Another Secure Boot bypass

[2025] DBX Inconsistency

- dbx is a crucial component of Secure Boot: it contains what **must not be trusted**
- Single source of truth for the entire ecosystem: UEFI Forum
- In July 2024, Microsoft publishes the DBX2024 update, blocking modules related to CVE-2024-28924 (Secure Boot Bypass)
- This update wasn't included in the UEFI Forum's dbx, so the update didn't propagate to non-MS devices (e.g. LVFS)
- For around 6 months, a Secure Boot bypass has been publicly known but not included in non-MS dbx

https://www.binarly.io/blog/from-trust-to-trouble-the-supply-chain-implications-of-a-broken-dbx

binarly

RSAC 2025 Conference

DEMO

Secure Boot bypass + Bootkit = ☠

Many Voices.
One Community.

RSAC 2025 Conference

AMD Microcode
Broken Signature Validation

[2025] EntrySign (CVE-2024-56161)

- Google researchers found an AMD microcode vulnerability that allows crafting valid microcode updates
- The microcode controls the low-level operations of the CPU:
  - Allows to override any CPU instruction (rdrand always returns 4)
  - Very difficult to detect, it basically infects the CPU
- Root cause: *"We noticed that the key from an old Zen 1 CPU was the example key of the NIST SP 800-38B publication and was reused until at least Zen 4 CPUs".*

https://bughunters.google.com/blog/5424842357473280/zen-and-the-art-of-microcode-hacking
https://www.binarly.io/blog/binarly-tracking-updates-for-cve-2024-56161-a-high-risk-microcode-flaw-in-amd-cpus

binarly

RSAC | 2025 Conference

Post Quantum Readiness
Device Security Implications

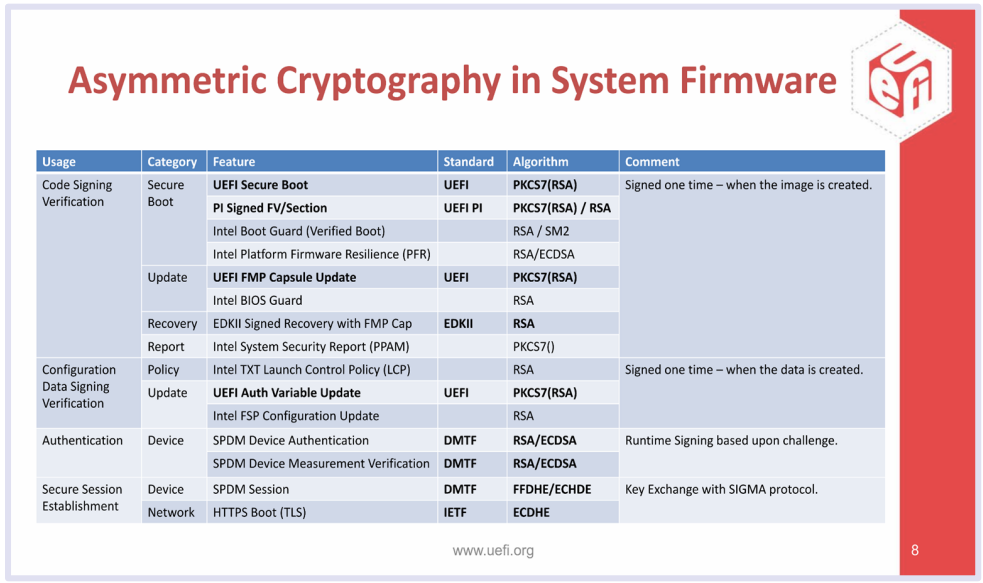
Post-Quantum Readiness

*"The migration will take time and will be more complex than people think. This is actually the driver. Even though **7–10 years sounds a long time** away, in reality the extent of the work needed might mean you are already too late."*

Phil Venables, CISO @ Google Cloud

https://www.philvenables.com/post/post-quantum-cryptography-migration-time-to-get-going

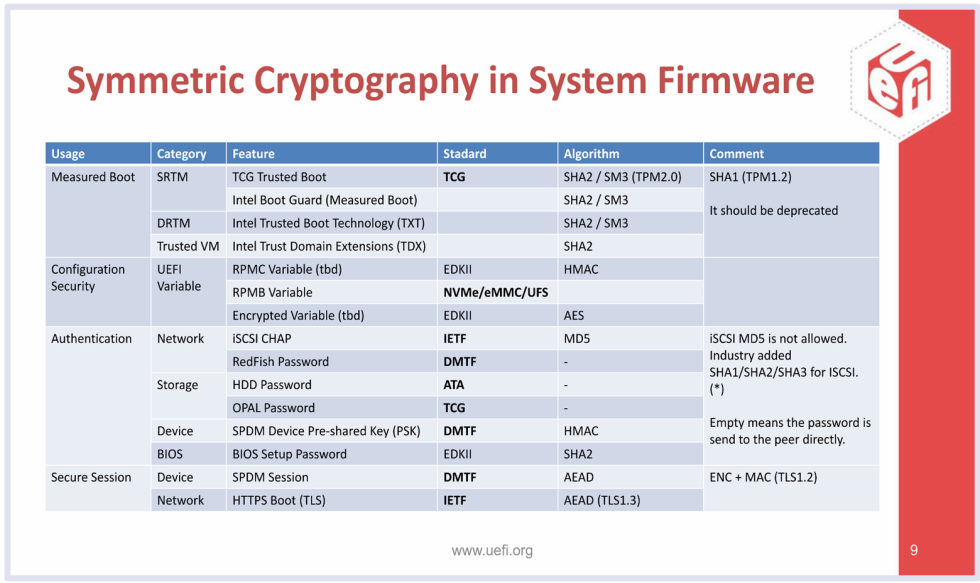
Post-Quantum Readiness in UEFI



Asymmetric Cryptography in System Firmware

Usage	Category	Feature	Standard	Algorithm	Comment
Code Signing Verification	Secure Boot	**UEFI Secure Boot**	**UEFI**	**PKCS7(RSA)**	Signed one time – when the image is created.
		PI Signed FV/Section	**UEFI PI**	**PKCS7(RSA) / RSA**	
		Intel Boot Guard (Verified Boot)		RSA / SM2	
		Intel Platform Firmware Resilience (PFR)		RSA/ECDSA	
	Update	**UEFI FMP Capsule Update**	**UEFI**	**PKCS7(RSA)**	
		Intel BIOS Guard		RSA	
	Recovery	EDKII Signed Recovery with FMP Cap	**EDKII**	**RSA**	
	Report	Intel System Security Report (PPAM)		PKCS7()	
Configuration Data Signing Verification	Policy	Intel TXT Launch Control Policy (LCP)		RSA	Signed one time – when the data is created.
	Update	**UEFI Auth Variable Update**	**UEFI**	**PKCS7(RSA)**	
		Intel FSP Configuration Update		RSA	
Authentication	Device	SPDM Device Authentication	**DMTF**	**RSA/ECDSA**	Runtime Signing based upon challenge.
		SPDM Device Measurement Verification	**DMTF**	**RSA/ECDSA**	
Secure Session Establishment	Device	SPDM Session	**DMTF**	**FFDHE/ECHDE**	Key Exchange with SIGMA protocol.
	Network	HTTPS Boot (TLS)	**IETF**	**ECDHE**	

www.uefi.org 8



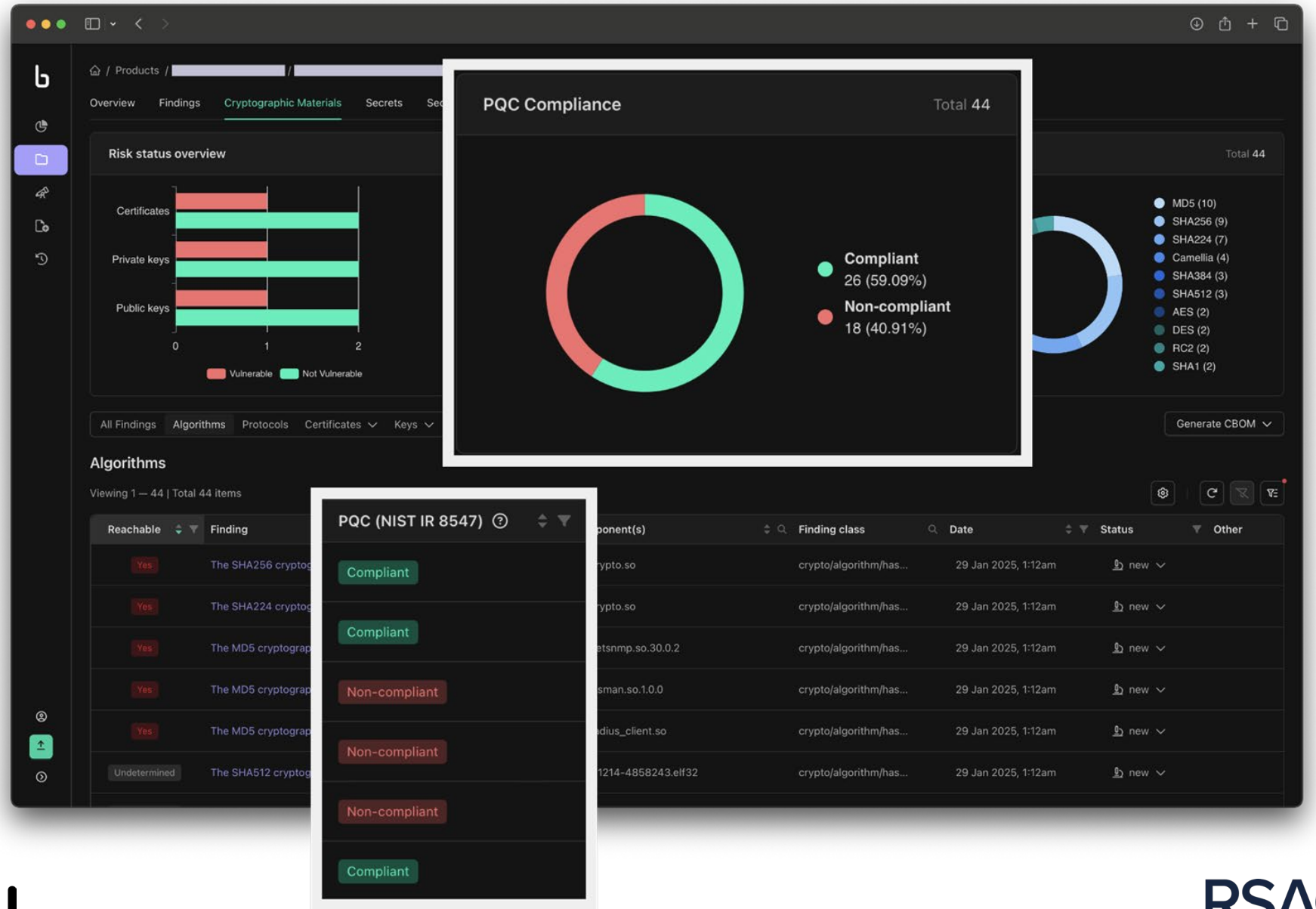
Symmetric Cryptography in System Firmware

Usage	Category	Feature	Stadard	Algorithm	Comment
Measured Boot	SRTM	TCG Trusted Boot	**TCG**	SHA2 / SM3 (TPM2.0)	SHA1 (TPM1.2)
		Intel Boot Guard (Measured Boot)		SHA2 / SM3	It should be deprecated
	DRTM	Intel Trusted Boot Technology (TXT)		SHA2 / SM3	
	Trusted VM	Intel Trust Domain Extensions (TDX)		SHA2	
Configuration Security	UEFI Variable	RPMC Variable (tbd)	EDKII	HMAC	
		RPMB Variable	**NVMe/eMMC/UFS**		
		Encrypted Variable (tbd)	EDKII	AES	
Authentication	Network	iSCSI CHAP	**IETF**	MD5	iSCSI MD5 is not allowed. Industry added SHA1/SHA2/SHA3 for ISCSI. (*)
		RedFish Password	**DMTF**	-	
	Storage	HDD Password	**ATA**	-	
		OPAL Password	**TCG**	-	Empty means the password is send to the peer directly.
	Device	SPDM Device Pre-shared Key (PSK)	**DMTF**	HMAC	
	BIOS	BIOS Setup Password	EDKII	SHA2	
Secure Session	Device	SPDM Session	**DMTF**	AEAD	ENC + MAC (TLS1.2)
	Network	HTTPS Boot (TLS)	**IETF**	AEAD (TLS1.3)	

www.uefi.org 9

- Ongoing discussion and few proof-of-concepts
- It will take years to update every component (huge complexity in firmware)

Source: *Post Quantum Cryptography impact to the UEFI Firmware,* UEFI 2021 Virtual Plugfest


Products
Overview
Findings
Cryptographic Materials
Secrets
Risk status overview
Certificates
Private keys
Public keys
Vulnerable
Not Vulnerable
PQC Compliance
Total 44
Compliant
26 (59.09%)
Non-compliant
18 (40.91%)
MD5 (10)
SHA256 (9)
SHA224 (7)
Camellia (4)
SHA384 (3)
SHA512 (3)
AES (2)
DES (2)
RC2 (2)
SHA1 (2)
All Findings
Algorithms
Protocols
Certificates
Keys
Generate CBOM
Algorithms
Viewing 1 — 44 | Total 44 items
Reachable
Finding
PQC (NIST IR 8547)
Finding class
Date
Status
Other
Compliant
Compliant
Non-compliant
Non-compliant
Non-compliant
Compliant
29 Jan 2025, 1:12am
new

RSAC™ | 2025 Conference

All this has happened before.
All this will happen again.

Apply

- Make sure your UEFI firmware is always up-to-date
  - Bad news: that's one of the few things you can actually do
  - *Demand* better security practices from your UEFI firmware vendors
- Hope the vendor cares about security
  - Many don't, especially for older devices
  - Some can be opaque about firmware issues (especially unfixable ones!)
- Understand below-the-os OS security defenses (e.g. Secure Boot)
  - …and their limitations

Summary

- The UEFI firmware ecosystem has been affected by the leak of many private keys
- The intricate UEFI supply-chain exacerbates this problem
  - Keys leaked from vendor A can be deployed on devices from vendor B
- Poor cryptographic key management
  - Test keys intended for development end up in real devices
  - Private keys stored unencrypted or encrypted with weak and easily guessable passwords

RSAC™ | 2025 Conference

Thank you!

binarly

Many Voices.
One Community.